

# $k$ -Anonymously Private Search over Encrypted Data

Shiyu Ji, Kun Wan  
 {shiyu, kun}@cs.ucsb.edu

## Abstract

In this paper we compare the performance of various homomorphic encryption methods on a private search scheme that can achieve  $k$ -anonymity privacy. To make our benchmarking fair, we use open sourced cryptographic libraries which are written by experts and well scrutinized. We find that Goldwasser-Micali encryption achieves good enough performance for practical use, whereas fully homomorphic encryptions are much slower than partial ones like Goldwasser-Micali and Paillier.

## 1 Introduction

How to securely and privately search over encrypted dataset has been popular in research community for the past decade [27, 28, 32, 33, 11, 10, 1]. A private search scheme should allow the client to have access to one of the posting lists in the inverted index, while the server hosting the index cannot tell which posting is accessed by the client. Also for privacy concern, the client should not know any other posting except the legally requested one. To address this problem, one possible way is to leverage Homomorphic Encryption [21, 22, 37, 38, 19, 4, 5]. Recently there is some debate over the question whether such a homomorphic encryption based private search scheme can be practically used [40, 36, 1]. We note that it is important which encryption method is used, since the involved operations may be very different, e.g., there are many modular exponential operations in Paillier [37, 38], while Goldwasser-Micali only requires quadratic exponentiation; some lattice based HEs (e.g. [6, 4]) require noise management techniques such as bootstrapping [16], re-linearization [6] and approximate eigenvector [20], which need more time. It will be interesting to see which HE method outperforms others in performance.

The main objective of this paper is to compare the performance on private search given by existing homomorphic encryption methods. The main challenge is that to write a cryptographic library is time-consuming and error-prone. Thus to do a fair comparison, we need well written libraries which are publicly available and have been scrutinized by experts. Fortunately there are many published libraries available for Homomorphic Encryption [26, 24, 23, 12], and many of them are written by the original paper authors [23, 12]. We will do a benchmark for each library and compare the performance results.

## 2 Preliminaries and Related Works

### 2.1 Homomorphic Encryption

Homomorphic encryption (HE) is an encryption system that allows some computations over ciphertexts without knowing the plaintexts or secret keys. A homomorphic encryption that allows the computation of any circuit (including additions and multiplications) over ciphertexts is called Fully Homomorphic Encryption (FHE). If the computation scope is restricted (e.g., only addition or multiplication by some constant), it is called Partially Homomorphic Encryption (PHE).

- *Partially Homomorphic Encryption:* There is a line of research works that can support HE with any times of additions: ElGamal [14], Goldwasser-Micali [21, 22], Benaloh [2], Paillier [37, 38], etc. A PHE  $\text{Enc}$  supporting addition can compute  $\text{Enc}(m_1 + m_2)$  only given the ciphertexts  $\text{Enc}(m_1)$  and  $\text{Enc}(m_2)$ .

Furthermore, Goldwasser-Micali can support the multiplication of two bits, i.e., to compute  $\text{Enc}(b_1 b_2)$  only given  $\text{Enc}(b_1)$  and  $\text{Enc}(b_2)$  where  $b_1, b_2 \in \{0, 1\}$ . Paillier can support multiplication by constant, i.e., to compute  $\text{Enc}(m_1 m_2)$  only given  $\text{Enc}(m_1)$  and  $m_2$  where  $m_1, m_2$  are plaintexts. However, PHE *cannot* support addition and multiplication over ciphertext simultaneously.

- *Fully Homomorphic Encryption*: Since the first time to be proposed by Craig Gentry [16], there is a long line of research works on FHE [45, 41, 42, 15, 18, 17, 6, 4, 3, 20, 5, 7]. So far most FHE schemes are built on difficult lattice problems [39]. By FHE, one may arbitrarily compute over ciphertexts, e.g., add and multiply for many times. The major challenge in the design of FHE is noise management, i.e., the noise in ciphertext grows rapidly when multiplying [6, 7]. Thus some time-consuming techniques like squashing [16, 15, 18, 17, 13], bootstrapping [16], re-linearization [6, 7], approximate eigenvector [20] are needed. The difficulty of noise management makes it very challenging to improve the performance of FHE in practice, and there is a long way to go to make an efficient FHE implementation by industry standard [35].

## 2.2 Private Information Retrieval and Private Search

A private information retrieval (PIR) scheme allows the client to obtain a posting in the inverted index from the server, while the server cannot learn which posting is requested by the client. Clearly the most trivial way is to give all the postings to the client. This takes too much bandwidth, and the client learns everything: if it is a dataset of medical records, the client learns the private information from anyone else. Hence a private search scheme further requires that the client cannot learn any other information except the requested posting.

There is a line of research on PIR [27, 9, 28, 32, 33]. These works tried to reduce the computation time and communication overhead of PIR by assuming some computational intractability, e.g., quadratic residuosity problem [21, 22], learning with errors [39], etc. So far the best result given by [33] is that the computation time is  $O(n/\log n)$  and the communication overhead is  $O(\log^2 n)$  where  $n$  is the size of dataset and the client requests one bit in the dataset.

Note that a PIR scheme that has absolutely no leakage is widely considered to be inefficient in practice (e.g., the response time can be 10 to 1000 seconds) [40, 36]. However, if we allow some acceptable leakage, can we obtain an efficient PIR scheme that can be used in the real world? In this paper we investigate this possibility.

## 2.3 $k$ -Anonymity Privacy

As a standard notion of privacy, together with its variants [34, 31],  $k$ -anonymity has been researched for many years [44, 43, 29, 30, 34, 31]. A key-value based dataset has  $k$ -anonymity if the value for each key cannot be distinguished from at least  $k - 1$  other values whose keys also appear in the dataset. For a  $k$ -anonymous search scheme, for each request from the client, there always exist  $k$  entries in the inverted index such that the server cannot tell which one of these  $k$  entries is requested by the client. There is absolutely no leakage if  $k$  equals the number of entries in the inverted index.

## 3 A Private Search Scheme Achieving $k$ -Anonymity

In this paper we investigate the performance of the private search scheme given as Algorithm 1. The HE based idea of this scheme is similar to [27, 28, 32, 33, 1].

Clearly the computation time of the scheme as above is  $O(kp)$  and the communication overhead is  $O(k + p)$ . The correctness can be verified as follows.

$$\text{Dec}_{\text{sk}}(e_s) = \text{Dec}_{\text{sk}}\left(\sum_{j=1}^k \text{Enc}_{\text{pk}}(b_j) \cdot D_j[s]\right) = \text{Dec}_{\text{sk}}\left(\text{Enc}_{\text{pk}}\left(\sum_{j=1}^k b_j \cdot D_j[s]\right)\right) = \text{Dec}_{\text{sk}}(\text{Enc}_{\text{pk}}(D_i[s])) = D_i[s].$$

---

**Algorithm 1:** A  $k$ -anonymous private search scheme.

---

- Data:**  $D$ : the inverted index containing  $k$  posting lists:  $D_1, \dots, D_k$ ;  
 $i$ : the index of the posting that the client requests;  $p$ : the length of posting list, i.e.,  $p = |D_j|$  for any index  $j$ ;  
**Enc:** a PHE supporting addition and multiplication by constant with public key  $\text{pk}$  and secret key  $\text{sk}$ .  
**Result:** The client learns  $D_i$  and nothing more, while the server cannot tell  $i$ .
- 1 For each index  $j \in [1, k]$ , the client computes  $c_j = \text{Enc}_{\text{pk}}(b_j)$  where  $b_j$  equals to 1 if  $i = j$  and 0 otherwise;
  - 2 The client sends  $C = \{c_1, \dots, c_k\}$  to the server;
  - 3 For each offset  $s \in [1, p]$ , the server computes  $e_s = \sum_{j=1}^k c_j \cdot D_j[s]$ , where  $D_j[s]$  denotes the  $s$ -th bit in the posting  $D_j$ ;
  - 4 The server sends  $E = \{e_1, \dots, e_p\}$  to the client;
  - 5 The client decrypts each ciphertext  $e_s$  in  $E$  by using  $\text{sk}$  to recover the posting list  $D_i$ ;
- 

For practical use, very often we do not need too large  $k$  (e.g., at most 100). However, the number of entries in the inverted index can be much larger. One possible solution is to divide the inverted index into several blocks with size  $k$ , and the client lets the server know which block will be requested. A leakage immediately follows since now the server knows which block is frequently requested, and thus the search pattern is not completely hidden. But if we only require  $k$ -anonymous, then the postings within one block are  $k$ -anonymous since the server cannot tell which one of them is requested.

## 4 Evaluation Results

In this section we summarize the evaluation results. By running similarly implemented testing code, we run the homomorphic encryption based private search schemes using the libraries as follows:

- Praveen Kumar’s C library *libgm* [26] implementing the Goldwasser-Micali cryptosystem.
- Gustavo Brunoro’s Python code [8] implementing the Goldwasser-Micali cryptosystem. We denote this code as PyGM.
- Mike Ivanov’s Python library [24] implementing the Paillier cryptosystem.
- Our Python code based on [8] implementing the Cachin-Micali scheme [9].<sup>1</sup>
- Shai Halevi and Victor Shoup’s HElib [23] (written in C++) implementing the Brakerski-Gentry-Vaikuntanathan (BGV) scheme [4, 5].
- Leo Ducas and Daniele Micciancio’s FHEW [12] (written in C++) implementing their own FHE scheme [13].

For the schemes which need large primes, we generate 2048-bit primes, which are considered to be secure so far [25]. For the lattice or LWE based FHE schemes, we use the minimum security configuration given by the authors. We choose the posting length to be 720 bits. We consider different  $k$ ’s in  $k$ -anonymity when evaluating.

We run the experiments on Linux Ubuntu 16.04 servers with 8 cores of 2.4 GHz AMD FX8320, 16GB memory. Since some libraries do not provide enough interfaces (e.g., except for encryption and decryption, FHEW only provides ciphertext evaluation for NAND to use), we cannot guarantee that all of our implementation are optimal.

---

<sup>1</sup>Cachin-Micali scheme is a PIR scheme, *not* a HE system.

libgm	PyGM	Paillier	Cachin-Micali	HElib	FHEW
0.84	0.015	0.024	0.40	6.13	14.8

Table 1: Key generation time (sec) for different  $k$ 's.

$k$	libgm	PyGM	Paillier	Cachin-Micali	HElib	FHEW
10	0.02	0.0002	0.93	0.40	1.03	6.8e-5
20	0.03	0.0004	1.86	0.57	2.04	1.3e-4
50	0.07	0.0007	4.59	1.59	5.18	3.3e-4
100	0.15	0.0012	19.90	3.02	15.76	6.7e-4

Table 2: Query encryption time (sec) for different  $k$ 's.

#### 4.1 Key Generation Time

Table 1 gives the key generation time results. Goldwasser-Micali and Paillier are the best. Note that libgm uses GNU MP Bignum library, while PyGM uses Miller-Rabin to test primality. GNU MP generates large primes that are safer to use, but takes more time.

#### 4.2 Query Encryption Time

Table 2 gives the query encryption time results. Goldwasser-Micali and FHEW achieve the best performance.

#### 4.3 Query Execution Time

The query execution includes the time that the server uses to generate the ciphertext to return based on the query, and the decryption time on the client side.

Table 3 gives the query execution time results. Goldwasser-Micali scheme achieves the best performance. Note that FHE schemes (HElib and FHEW) take more time on the additions and multiplications over ciphertexts, since the noise in ciphertexts grows rapidly by multiplication and hence the scheme needs to control the noise.

#### 4.4 Communication Overhead

Table 4 gives the communication overhead results. We evaluate communication overhead as the number of bytes communicated between the client and server during the private search protocol. Goldwasser-Micali, Cachin-Micali and Paillier achieve the best performance, and their overhead grows slowly as  $k$  increases.

### 5 Conclusion and Future Works

Based on the benchmarking results, Goldwasser-Micali encryption achieves good enough performance for practical use, whereas fully homomorphic encryptions are much slower than partial ones like Goldwasser-

$k$	libgm	PyGM	Paillier	Cachin-Micali	HElib	FHEW
10	0.172	0.113	0.307	0.43	1.61	16k
20	0.175	0.130	0.316	0.81	3.08	33k
50	0.185	0.164	0.361	1.87	7.70	82k
100	0.214	0.232	0.440	6.09	20.93	164k

Table 3: Query execution time (sec) for different  $k$ 's.

$k$	libgm	PyGM	Paillier	Cachin-Micali	HElib	FHEW
10	11680	6440	6440	6528	1408	22k
20	11840	6512	6512	6600	2688	42k
50	12320	6776	6776	6864	6528	102k
100	13120	7160	7160	7248	12928	202k

Table 4: Communication overhead (bytes) for different  $k$ 's.

Micali and Paillier. We suggest that it may be worthwhile to research on the possibility of incorporating Goldwasser-Micali method into private search scheme in the future works.

## References

- [1] Carlos Aguilar-Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. Xpir: Private information retrieval for everyone. *Proceedings on Privacy Enhancing Technologies*, 2:155–174, 2016.
- [2] Josh Benaloh. Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128, 1994.
- [3] Joppe W Bos, Kristin E Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In *IMA Int. Conf.*, pages 45–64. Springer, 2013.
- [4] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.
- [5] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13, 2014.
- [6] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Annual cryptology conference*, pages 505–524. Springer, 2011.
- [7] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.
- [8] Gustavo Brunoro. Python code for goldwasser micali scheme. <https://gist.github.com/brunoro/5893701/>.
- [9] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 402–414. Springer, 1999.
- [10] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In *NDSS*, volume 14, pages 23–26. Citeseer, 2014.
- [11] David Cash, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, Marcel-Cătălin Roşu, and Michael Steiner. Highly-scalable searchable symmetric encryption with support for boolean queries. In *Advances in Cryptology-CRYPTO 2013*, pages 353–373. Springer, 2013.
- [12] Léo Ducas and Daniele Micciancio. Fhew. <https://github.com/lucas/FHEW/>.
- [13] Léo Ducas and Daniele Micciancio. Fhew: Bootstrapping homomorphic encryption in less than a second. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640. Springer, 2015.

- [14] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.
- [15] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *Annual Cryptology Conference*, pages 116–137. Springer, 2010.
- [16] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
- [17] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 107–109. IEEE, 2011.
- [18] Craig Gentry and Shai Halevi. Implementing gentrys fully-homomorphic encryption scheme. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 129–148. Springer, 2011.
- [19] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology-CRYPTO 2013*, pages 75–92. Springer, 2013.
- [20] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology-CRYPTO 2013*, pages 75–92. Springer, 2013.
- [21] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377. ACM, 1982.
- [22] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.
- [23] Shai Halevi and Victor Shoup. Helib. <https://github.com/shaih/HElib>.
- [24] Mike Ivanov. paillier. <https://github.com/mikeivanov/paillier/>.
- [25] Burt Kaliski. Twirl and rsa key size, 2003.
- [26] Praveen Kumar. libnum. <https://github.com/prvnkumar/libgm/>.
- [27] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 364–373. IEEE, 1997.
- [28] Eyal Kushilevitz and Rafail Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 104–121. Springer, 2000.
- [29] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.
- [30] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE, 2006.
- [31] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

- [32] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *International Conference on Information Security*, pages 314–328. Springer, 2005.
- [33] Helger Lipmaa. First cpir protocol with data-dependent computation. In *International Conference on Information Security and Cryptology*, pages 193–210. Springer, 2009.
- [34] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramanian. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.
- [35] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.
- [36] Femi Olumofin and Ian Goldberg. Revisiting the computational practicality of private information retrieval. In *International Conference on Financial Cryptography and Data Security*, pages 158–172. Springer, 2011.
- [37] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.
- [38] Pascal Paillier and David Pointcheval. Efficient public-key cryptosystems provably secure against active adversaries. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 165–179. Springer, 1999.
- [39] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.
- [40] Radu Sion and Bogdan Carbunar. On the computational practicality of private information retrieval. In *Proceedings of the Network and Distributed Systems Security Symposium*, pages 2006–06. Internet Society, 2007.
- [41] Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography*, pages 420–443. Springer, 2010.
- [42] Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 377–394. Springer, 2010.
- [43] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.
- [44] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
- [45] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer, 2010.